AWS DevOps Linux Commands Basics

Phase-1:

1. ip command
2. ping
3. ls or ls -l or ls -al
4. touch or touch file{1..10}
5. pwd
6. mkdir or mkdir -p
7. cd or cd .. or cd -
8. rm or rm -rf
9. rmdir
10. date
11. w
12. whoami
13. uptime
14. hostname
15. cat or cat > <filename>
16. cp command
17. mv command
18. locate --> mlocate package
19. sudo
20. df command
21. du command
22. lsblk
23. sleep
24. jobs
25. fg and bg commands
26. uname
27. wget
28. history
29. wc
30. echo or echo $(command) or echo {1..5}
31. man command
32. history

33. free
34. top
35. ps
36. env or printenv
37. alias
38. less
39. diff
40. uniq or uniq -d or uniq -c (with help of sort command)
41. sort
42. type (TODO)
43. which
44. nohup --> to run command even after remote connection disconnected or prevent command to be halted

Phase-2:

1. apt list --upgradable --> list all packages which are ready to upgrade
2. apt-mark hold packagename or echo "packagename hold" | sudo dpkg --set-selections or aptitude hold packagename
3. apt-mark showhold or dpkg --get-selections
4. distro-info (to list distribution information)
5. disown (to remove jobs)
6. How to perform dist upgrade from one LTS to another LTS version (20.04 to 22.04 LTS for example)
    1. /etc/update-manager/release-upgrades --> make sure it was set to lts
    2. apt-mark showhold to list all holded packages
    3. if we see any package on hold, unhold then using #apt-mark unhold package1 package2
    4. Now run #apt update and #apt upgrade
    5. do-release-upgrade -d
7. snap tool
8. apport-cli --> to report bugs
9. Kernel Crash Dump --> https://ubuntu.com/server/docs/kernel-crash-dump

```
APT::Periodic::Update-Package-Lists "0";
APT::Periodic::Download-Upgradeable-Packages "0";
```

```
APT::Periodic::AutocleanInterval "0"; -->  The option accepts days so of course it is 30.
1 would be arbitrary if it would not be 1 day since it could be 1 day, 1 week, 1 month, 1
year, 1 decade, 1 century.
APT::Periodic::Unattended-Upgrade "1";
```

Reference:
https://debian-handbook.info/browse/stable/sect.regular-upgrades.html

Disk names that are assigned in Linux OS:

Type of disk	Disk names	Commonly used disk names
IDE	/dev/hd[a-h]	/dev/hda, /dev/hdb
SCSI	/dev/sd[a-p]	/dev/sda, /dev/sdb
ESDI	/dev/ed[a-d]	/dev/eda
XT	/dev/xd[ab]	/dev/xda

Important commands for disk/partitions

```
lsblk
lsblk -fs
lsblk -o NAME,SIZE,OWNER,GROUP,MODE
```

How to check filesystem error/issues on a disk/partition?

```
btrfs check --force /dev/sdb
e2fsck /dev/sdb or e2fsck /dev/sdb1
To auto repair all issues without prompting for conformation --> e2fsck -p [-y] /dev/sdb
Report filesystem issue, buit do not fix it --> e2fsck -n /dev/sdb[1]
```

Filesystem Table entries in /etc/fstab

Block device	Mountpoint	Filesystem type	Mount options	Filesystem dump	Filesystem check order

Block devices under /dev directory Example: /dev/sda[1,2,3] LABEL UUID (Universal Unique Identifier) UUID and LABEL check be checked with lsblk -fs and blkid commands	Target directory to which Block device has be attached to store the data Example: /mnt /sudheer	Filesystem that was used on disk or partition has to be specified Example: ext4 btrfs xfs nfs	Default mount options **rw/ro** (read-write or read-only) **suid** (setuid and setgid bits) **dev** (interpret characters and block devices on the filesystem) **exec/noexec** (allow executing binaries and scripts) **auto/noauto** (mount the filesystem when the -a option of the mount commands is used while boot) **nouser/user** (make the filesystem not mountable by the standard user) **async** (perform I/O operations on the filesystem asynchronously)	Value could be 0 or 1 To initiate backup of filesystem dump (if package "dump" installed for ext2/3/4) But this is deprecated, no longer needed	On boot if filesystem has to be checked on the disk or partition 0 --> disable 1 --> always to be assigned to root 2 --> to enable check for other mounts

Difference between ext4 and btrfs filesystem

ext4	**btrfs**	**Comments**

Extended version 4 Filesystem	B-Tree Filesystem	The Ext4 filesystem has journaling support. So, your files should be safe even when there's a power failure. It's a good filesystem for everyday use.
Journaled Filesystem	Modern COW (Copy-on-Write) Filesystem	
Partition size <= 1EiB	Partition size <= 16 EiB	
File size <= 16 TiB	File size <= 16 EiB	
Length of Filename 255 characters (255 bytes)	Length of Filename 255 characters (255 bytes)	
Max files creation 232 files (= 4,294,967,296 ~= 4 billion)	264 files (= 18,446,744,073,709,551,616 ~= 18 quintillion)	
Snapshot of Filesystem is not supported	It supports Filesystem snapshots	Filesystem snapshot is an important feature. Using this feature, you can take a snapshot of your filesystem before trying out anything risky. If things do not go as planned, you can go back to an early state where everything worked. This is a built-in feature of the Btrfs filesystem. You don't need any 3rd-party tools/software to do that on a Btrfs filesystem.

Supports Filesystem level encryption	It doesn't support Filesystem level encryption	
Filesystem level duplication is not supported	It supports Filesystem level duplication	Duplication is a technique to eliminate/remove duplicate copies of data from the filesystem and keep only one copy of data (unique data) on the filesystem. This technique is used to save disk spaces.
Ext4 filesystem does not support multiple devices	Btrfs filesystem supports multiple devices and has built-in RAID support	The Ext4 filesystem does not support multiple devices. You can’t span a single Ext4 filesystem over multiple disks or partitions. To combine multiple storage devices and partitions in an Ext4 filesystem, you have to use 3rd-party logical volume managers like **LVM 2**. To set up RAID, you have to use 3rd-party tools like DM-RAID or MDADM.
Filesystem level compression: does not support	It is supported	It can compress a single directory or a single file or the entire filesystem to save disk space.

How to remove a filesystem assigned on a disk/partition?

```
root@ubuntudockerserver:~# lsblk -fs
NAME                 FSTYPE      LABEL UUID                                   FSAVAIL
FSUSE% MOUNTPOINT
```

```
loop0                 squashfs                                                  0
100% /snap/core18/2344
loop1                 squashfs                                                  0
100% /snap/helm/353
loop2                 squashfs                                                  0
100% /snap/dotnet-sdk/162
loop3                 squashfs                                                  0
100% /snap/core18/2409
loop4                 squashfs                                                  0
100% /snap/core20/1434
loop5                 squashfs                                                  0
100% /snap/snapd/15904
loop6                 squashfs                                                  0
100% /snap/dotnet-sdk/168
loop7                 squashfs                                                  0
100% /snap/core20/1494
loop8                 squashfs                                                  0
100% /snap/kubectl/2419
loop9                 squashfs                                                  0
100% /snap/snapd/15534
loop10                squashfs                                                  0
100% /snap/lxd/22526
loop11                squashfs                                                  0
100% /snap/kubectl/2433
loop12                squashfs                                                  0
100% /snap/lxd/22753
sda1
└─sda
sda2                  ext4              3f07f99b-877c-40ee-8bff-5f72cbfd5b9f    702.5M
21% /boot
└─sda
sdb
sr0
ubuntu--vg-ubuntu--lv ext4              1d39120d-bb69-47b2-83cf-ba5d2e1d1025     18.4G
57% /
└─sda3                LVM2_member       rhcex9-5Zof-1Hea-EJXZ-n7Ti-DXs1-CONAqY
  └─sda
root@ubuntudockerserver:~# mkfs.ext4 /dev/sdb
mke2fs 1.45.5 (07-Jan-2020)
Found a dos partition table in /dev/sdb
Proceed anyway? (y,N) N
root@ubuntudockerserver:~# wipefs /dev/sdb
DEVICE OFFSET TYPE UUID LABEL
sdb    0x1fe  dos
root@ubuntudockerserver:~# wipefs -a /dev/sdb
/dev/sdb: 2 bytes were erased at offset 0x000001fe (dos): 55 aa
/dev/sdb: calling ioctl to re-read partition table: Success
```

```
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# mkfs.ext4 /dev/sdb
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 20971520 4k blocks and 5242880 inodes
Filesystem UUID: 27245806-153f-470f-8631-2f3e2f287ae7
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000
Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# wipefs /dev/sdb
DEVICE OFFSET TYPE UUID                                 LABEL
sdb    0x438  ext4 27245806-153f-470f-8631-2f3e2f287ae7
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# wipefs /dev/sda*
DEVICE OFFSET       TYPE        UUID                                   LABEL
sda    0x200        gpt
sda    0x18fffffe00 gpt
sda    0x1fe        PMBR
sda2   0x438        ext4        3f07f99b-877c-40ee-8bff-5f72cbfd5b9f
sda3   0x218        LVM2_member rhcex9-5Zof-1Hea-EJXZ-n7Ti-DXs1-CONAqY
root@ubuntudockerserver:~#
```

How to recover the data?

```
root@ubuntudockerserver:~# lsblk -fs /dev/sdb*
NAME FSTYPE LABEL UUID FSAVAIL FSUSE% MOUNTPOINT
sdb
root@ubuntudockerserver:~# mkfs.btrfs /dev/sdb
btrfs-progs v5.4.1
See http://btrfs.wiki.kernel.org for more information.
Label:              (null)
UUID:               14ee7168-f47e-4804-b612-946be2e0d5b6
Node size:          16384
Sector size:        4096
Filesystem size:    80.00GiB
Block group profiles:
  Data:             single            8.00MiB
  Metadata:         DUP               1.00GiB
  System:           DUP               8.00MiB
SSD detected:       no
Incompat features:  extref, skinny-metadata
Checksum:           crc32c
```

```
Number of devices:  1
Devices:
   ID        SIZE  PATH
    1    80.00GiB  /dev/sdb
root@ubuntudockerserver:~# mount /dev/sdb /mnt
root@ubuntudockerserver:~# df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb         80G  3.5M   78G   1% /mnt
root@ubuntudockerserver:~# fallocate -l 50G /mnt/sudheer_demo
root@ubuntudockerserver:~# df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb         80G   51G   28G  65% /mnt
root@ubuntudockerserver:~#

root@ubuntudockerserver:~# wipefs --all --backup /dev/sdb
wipefs: error: /dev/sdb: probing initialization failed: Device or resource busy
root@ubuntudockerserver:~# umount /mnt
root@ubuntudockerserver:~# df -h /mnt
Filesystem                         Size  Used Avail Use% Mounted on
/dev/mapper/ubuntu--vg-ubuntu--lv   49G   28G   19G  60% /
root@ubuntudockerserver:~# wipefs --all --backup /dev/sdb
/dev/sdb: 8 bytes were erased at offset 0x00010040 (btrfs): 5f 42 48 52 66 53 5f 4d
root@ubuntudockerserver:~# lsblk -fs /dev/sdb*
NAME FSTYPE LABEL UUID FSAVAIL FSUSE% MOUNTPOINT
sdb
root@ubuntudockerserver:~# ls -l wipefs-sdb-0x00010040.bak
-rw------- 1 root root 8 May 28 15:04 wipefs-sdb-0x00010040.bak
root@ubuntudockerserver:~# dd if=~/wipefs-sdb-0x00010040.bak of=/dev/sdb
seek=$((0x00010040)) bs=1 conv=notrunc
8+0 records in
8+0 records out
8 bytes copied, 0.00168053 s, 4.8 kB/s
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# lsblk -fs /dev/sdb*
NAME FSTYPE LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sdb  btrfs        14ee7168-f47e-4804-b612-946be2e0d5b6
root@ubuntudockerserver:~# mount /dev/sdb /mnt
root@ubuntudockerserver:~# df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb         80G   51G   28G  65% /mnt
root@ubuntudockerserver:~# ls -l /mnt/
total 52428800
-rw-r--r-- 1 root root 53687091200 May 28 15:03 sudheer_demo
root@ubuntudockerserver:~#
```

Reference: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ext4backup --> backup

filesystem

How to reset root user password for Ubuntu 20.04 LTS server?

1. Enter to grub mode
2. Go to line which starts with "Linux"
3. Remove from the word from "ro" and add "rw init=/bin/bash"
4. ctrl + x
5. check the root filesystem is mounted as read-write mode #mount | grep -w /
6. passwd command to set password for root user
7. exec /sbin/init
8. Now check the password
9. Optional: mount / -o remount,ro